

# Oracle Database 19c Easy Connect Plus

Configurable Database Connection Syntax

WHITE PAPER / DECEMBER 5, 2019

ORACLE®

## Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

## TABLE OF CONTENTS

Easy Connect Plus Overview.....	4
Easy Connect Background .....	4
Easy Connect Plus Syntax.....	4
Protocol .....	4
Hosts .....	5
Ports .....	5
Service Name.....	5
Server Type.....	5
Instance Name .....	5
Parameters.....	5
Parameters .....	6
Security Parameters.....	6
Proxy Parameters .....	6
Description Parameters.....	6
Examples .....	8
Conclusion .....	11

## EASY CONNECT PLUS OVERVIEW

Oracle Database 19c introduces a significantly enhanced connection string syntax called Easy Connect Plus.

By reducing the need for external configuration files, Easy Connect Plus makes it easier to use features such as TLS connections, wallets, load balancing, connection timeouts, and to tune network buffer sizes. For example, when accessing Oracle Database Cloud services, it is now easy to use TLS for secure communication by including the wallet location in the connection string.

Easy Connect Plus can be used in JDBC and .NET applications. It is also usable in applications and drivers built on the C language Oracle Call Interface (OCI) such as the Oracle Database drivers for Python, Node.js, PHP, Ruby, and Go.

## EASY CONNECT BACKGROUND

Easy Connect is a syntax currently used by Oracle programs and applications when connecting to an Oracle Database service. At its simplest, an Easy Connect string includes a host name, an optional port, and the database service name:

```
database_host[:port][/[service_name]
```

For example, when Oracle Database is running on mydbhost.example.com with the default Pluggable Database service "orclpdb1", then a typical connection in SQL\*Plus would be:

```
SQL> connect scott/pw@mydbhost.example.com/orclpdb1
```

Or in Python:

```
connection = cx_Oracle.connect('scott', password,  
                               'mydbhost.example.com/orclpdb1')
```

## EASY CONNECT PLUS SYNTAX

The Oracle Database 19c enhanced Easy Connect Plus syntax is a superset of Easy Connect that supports more connection options. The syntax is available in Oracle Database drivers (JDBC, ODP.Net, cx\_Oracle, node-oracledb etc) that use Oracle Client 19c and connect to Oracle Database 11.2 or later.

The Oracle Database 19c Easy Connect Plus syntax is:

```
[[protocol:][//]host1{,host2}[:port1]{,host2:port2}[/[service_name][:  
server_type][/instance_name]][?parameter_name=value{&parameter_name=  
value}]
```

### Protocol

The protocol specifies the transport protocol to be used while connecting to the database host. In 19c, the supported values of protocol are TCP and TCPS. The default is TCP.

## Hosts

The host list is a comma-separated list of host names or IP addresses which are used to connect to a database.

When multiple hosts are used, load balancing is turned on automatically.

## Ports

Port numbers indicate the ports that the database services are listening on. The default is 1521.

When using multiple hosts, if the same port is used on each host, then you can use:

```
host1,host2:port
```

Otherwise you can indicate different ports should be used:

```
host1:port1,host2:port2
```

## Service Name

The service name is the [service](#) the database is known by. It is a name comprised of the database name and domain name, entered during installation or database creation.

Service names can be found by running “lsnrctl services” on the database host.

Note that database “system identifiers” (SIDs) cannot be used in Easy Connect or Easy Connect Plus syntax.

## Server Type

The server type specifies what kind of server is used on the database host to handle the connection. It can be one of DEDICATED, SHARED, or POOLED. The default is to leave it unspecified and let the listener decide.

The common, traditional server type is [dedicated](#).

With [shared servers](#) the server processes are shared, but the session state is not shared. Session state is in the SGA.

A [pooled server](#) is one used by [Database Resident Connection Pooling](#) (DRCP). The server process and session state are both shared. Session state is in the PGA.

Shared servers and DRCP are used when the database host does not have enough memory to handle a large number of dedicated connections.

## Instance Name

If you need to connect to a specific instance that the service represents, you can use an instance name.

## Parameters

Parameters are name-value pairs that control the behavior of connections. The syntax uses ‘?’ to indicate start of parameters and a ‘&’ delimiter between each parameter. Leading and trailing white spaces are ignored within parameter values. If whitespace is required as part of the value, it should be placed within double quotes. Parameters are described in the next section.

## PARAMETERS

Easy Connect Plus allows Security, Proxy and Description parameters to be specified.

Any parameters not noted here are passed through as Description parameters, allowing further (or future) options to be used when connecting.

### Security Parameters

Security parameters are used to configure a TLS connection to a database. When using these parameters, the protocol must be set to **TCPS**.

#### SSL\_SERVER\_CERT\_DN

The distinguished name (DN) of the database server.

When **SSL\_SERVER\_DN\_MATCH** is **ON**, this DN is matched with the DN from the server certificate.

#### SSL\_SERVER\_DN\_MATCH

Specifies if server-side certification validation through distinguished name (DN) matching should be enforced. The value can be **ON** or **OFF**. The default value is **ON** (from Oracle Client 19.2).

This parameter is used in conjunction with **SSL\_SERVER\_CERT\_DN**.

From Oracle Client 19.3 onwards, if the **SSL\_SERVER\_CERT\_DN** parameter is not specified, then by default the database server hostname is checked against the Common Name (CN) and Server Alternate Name (SAN) fields from the server certificate. Connection will not succeed if the check fails.

#### WALLET\_LOCATION

Sets the directory containing Oracle wallets. Wallets are certificates, keys, and trustpoints processed by SSL.

Only filesystem wallets are supported with this method.

### Proxy Parameters

Proxy parameters configure traffic to be routed through a proxy server.

Successful connection depends on specific proxy configurations. The performance of data transfers depends on proxy capacity. Oracle does not recommend using this feature in production environments where performance is critical.

#### HTTPS\_PROXY

Sets an HTTP proxy host name or IP address for tunneling TLS client connections. Non-TLS connections cannot be tunneled.

#### HTTPS\_PROXY\_PORT

Sets an HTTP proxy host port for tunneling TLS client connections.

### Description Parameters

All other parameters will be description parameters. These include:

#### **CONNECT\_TIMEOUT**

Sets the timeout duration in seconds for an application to establish an Oracle Net connection. There is no timeout by default.

#### **EXPIRE\_TIME**

Sets a time interval in minutes to send probes to verify that connections are active. Setting a value greater than 0 ensures that connections are not left open indefinitely due to an abnormal database server termination. If **EXPIRE\_TIME** is used to prevent firewalls terminating idle connections, then the value should be just less than half of the firewall timeout period.

#### **FAILOVER**

Enables or disables connect-time failover for multiple hosts. The value can be **ON** or **OFF**. The default is **ON**.

#### **LOAD\_BALANCE**

Enables or disables Oracle client load balancing for multiple hosts. The value can be **ON** or **OFF**. The default is **ON**.

#### **RECV\_BUF\_SIZE**

Sets the TCP/TCPs socket receive buffer size in bytes. This should be tuned to accommodate the bandwidth-delay product (BDP) of the connection. The database server configuration file sqlnet.ora should also set this parameter.

Large buffer sizes help applications queue more data to the operating system and utilize available bandwidth.

A starting recommendation is to set the buffer size to the BDP. As an example, on a 40 Mbps network with 25 millisecond round-trip time between endpoints, the BDP is  $(40 * 1000 / 8) \text{ KB/second} * (0.025) \text{ seconds}$ , which is 125 KB. The **RECV\_BUF\_SIZE** parameter could be set to 125000.

#### **SDU**

Sets the Oracle Net Session Data Unit packet size in bytes. Larger **SDU** values give better network throughput due to fewer system calls and lower CPU usage, at the expense of memory. The database server configuration should also set this parameter.

For bulk data transfers, set this to 64 K.

#### **SEND\_BUF\_SIZE**

Sets the TCP/TCPs socket send buffer size in bytes. This should be tuned to accommodate the bandwidth-delay product (BDP) of the connection. The database server configuration file sqlnet.ora should also set this parameter.

Large buffer sizes help applications queue more data to the operating system and utilize available bandwidth.

A starting recommendation is to set the buffer size to the BDP, see **RECV\_BUF\_SIZE** for an example.

#### **SOURCE\_ROUTE**

Enables routing through multiple hosts. The value can be **ON** or **OFF**. The default is **OFF**.

#### **RETRY\_COUNT**

Sets the number of times the list of hosts is traversed when attempting to connect to Oracle Database.

Connection is attempted in turn to each host in the host list, until a connection is successful or the whole host list has been traversed the specified number of times. If only one host is used, then connection to this host is attempted the specified number of times.

#### **RETRY\_DELAY**

Sets the delay in seconds between retries of host list traversal. The default is 0. This parameter works in conjunction with the **RETRY\_COUNT** parameter.

#### **TRANSPORT\_CONNECT\_TIMEOUT**

Sets the transport connect timeout duration in seconds for a client to establish an Oracle Net connection to an Oracle database. This is a subset of **CONNECT\_TIMEOUT** and only applies to TCP and TCPS connection establishment.

Other parameters can be also be used. They are treated as parameters in the **DESCRIPTION** section of an equivalent tnsnames.ora connection descriptor entry. They are validated during connection.

## **EXAMPLES**

This section includes various examples of Easy Connect Plus syntax and how each string converts into a connect descriptor such as might be used in a tnsnames.ora configuration file.

### **Example 1 - Protocol**

```
tcp://salesserver1:1521/sales.us.example.com
```

converts into the following descriptor

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

### **Example 2 – Multiple hosts**

```
salesserver1,salesserver2:1782/sales.us.example.com
```

converts into the following descriptor

```
(DESCRIPTION=
  (LOAD_BALANCE=ON)
  (ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521)))
```



```
(ADDRESS=(PROTOCOL=tcp) (HOST=salesserver2) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

### Example 3 – Multiple hosts and ports

```
salesserver1:1521,salesserver2,salesserver3:1522/sales.us.example.com
```

converts into the following descriptor

```
(DESCRIPTION=
  (LOAD_BALANCE=ON)
  (ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
  (ADDRESS=(PROTOCOL=tcp) (HOST=salesserver2) (PORT=1522))
  (ADDRESS=(PROTOCOL=tcp) (HOST=salesserver3) (PORT=1522))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

### Example 4 – IPv6 address as the host

```
[2606:b400:41c:8a19:f816:3eff:fe8f:e3ef]:1521/sales.us.example.com
```

converts into the following descriptor

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=2606:b400:41c:8a19:f816:3eff:fe8f:e3ef)
  (PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

Note that a numeric IPv6 address has to be placed between enclosing square brackets.

### Example 5 – TLS Connections

```
tcps://salesserver1:1521/sales.us.example.com
```

converts into the following descriptor

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1521))
  (SECURITY=(SSL_SERVER_DN_MATCH=ON))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

Note that `SSL_SERVER_DN_MATCH=ON` gets automatically added to the full connect string in order to enforce server DN matching. The hostname used for connecting (salesserver1 in this example) is matched against the CN and SAN fields in the server certificate.

### Example 6 – TLS Connections

```
tcps://salesserver1:1521/sales.us.example.com?ssl_server_dn_match=off
```

converts into the following descriptor

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1521))
  (SECURITY=(SSL_SERVER_DN_MATCH=OFF))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

### Example 7 – Wallet location

```
tcps://salesserver1:1521/sales.us.example.com?wallet_location="/tmp/oracle"
```

converts into the following descriptor

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
  (SECURITY=(MY_WALLET_DIRECTORY=/tmp/oracle) (SSL_SERVER_DN_MATCH=ON))
)
```

### Example 8 – Distinguished Name Matching

```
tcps://salesserver1:1521/sales.us.example.com?ssl_server_cert_dn="cn=sales,cn=OracleContext,dc=us,dc=example,dc=com"
```

converts to the following descriptor

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1521))
  (SECURITY=(SSL_SERVER_DN_MATCH=ON)
  (SSL_SERVER_CERT_DN=cn=sales,cn=OracleContext,dc=us,dc=example,dc=com))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
)
```

### Example 9 – HTTPS Proxy

```
tcps://salesserver1:1521/sales.us.example.com?https_proxy=www-proxy.mycompany.com&https_proxy_port=80
```

converts into the following descriptor

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1521)
    (https_proxy=www-proxy.mycompany.com) (https_proxy_port=80))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
)
```

### Example 10 – Session Data Unit

```
salesserver1:1521/sales.us.example.com?sdu=16384
```

converts into the following descriptor

```
(DESCRIPTION=
  (SDU=16384)
  (ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
)
```

### Example 11 – Connect Timeouts

```
salesserver1:1521/sales.us.example.com?connect_timeout=60&transport_connect_timeout=30&retry_count=3
```

converts into the following descriptor

```
(DESCRIPTION=
  (RETRY_COUNT=3) (CONNECT_TIMEOUT=60) (TRANSPORT_CONNECT_TIMEOUT=30)
)
```

```
(ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))  
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

### Example 12 – Quoting Easy Connect Strings

Quoting of parameters may be required in some tools and environments.

The following example invokes SQL\*Plus 19c in a Linux shell. This illustrates the placement of single and double quotes to escape interpretation of special characters by the shell:

```
$ sqlplus  
scott@'tcps://salesserver1:1521/sales.us.example.com?ssl_server_cert  
_dn="cn=sales,cn=Oracle Context  
Server,dc=us,dc=example,dc=com"&sdu=8128&connect_timeout=60'
```

### CONCLUSION

Oracle Database 19c Easy Connect Plus syntax makes it easier to deploy highly available and secure applications without needing external configuration files for connection management.

## ORACLE CORPORATION

### Worldwide Headquarters

500 Oracle Parkway, Redwood Shores, CA 94065 USA

### Worldwide Inquiries

TELE + 1.650.506.7000 + 1.800.ORACLE1

FAX + 1.650.506.7200

oracle.com

## CONNECT WITH US

Call +1.800.ORACLE1 or visit [oracle.com](http://oracle.com). Outside North America, find your local office at [oracle.com/contact](http://oracle.com/contact).

 [blogs.oracle.com/oracle](http://blogs.oracle.com/oracle)

 [facebook.com/oracle](http://facebook.com/oracle)

 [twitter.com/oracle](http://twitter.com/oracle)

## Integrated Cloud Applications & Platform Services

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 1219

White Paper **Oracle Database 19c Easy Connect Plus** Configurable Database Connection Syntax Oracle Database 19c Easy Connect Plus Oracle Database 19c Easy Connect Plus  
December 2019 December 2019  
Author: [OPTIONAL]  
Contributing Authors: [OPTIONAL]



Oracle is committed to developing practices and products that help protect the environment

ORACLE®